PC Serial Peripheral Design (I)

Part 1: Introduction, course hardware, first program

By B. Kainka

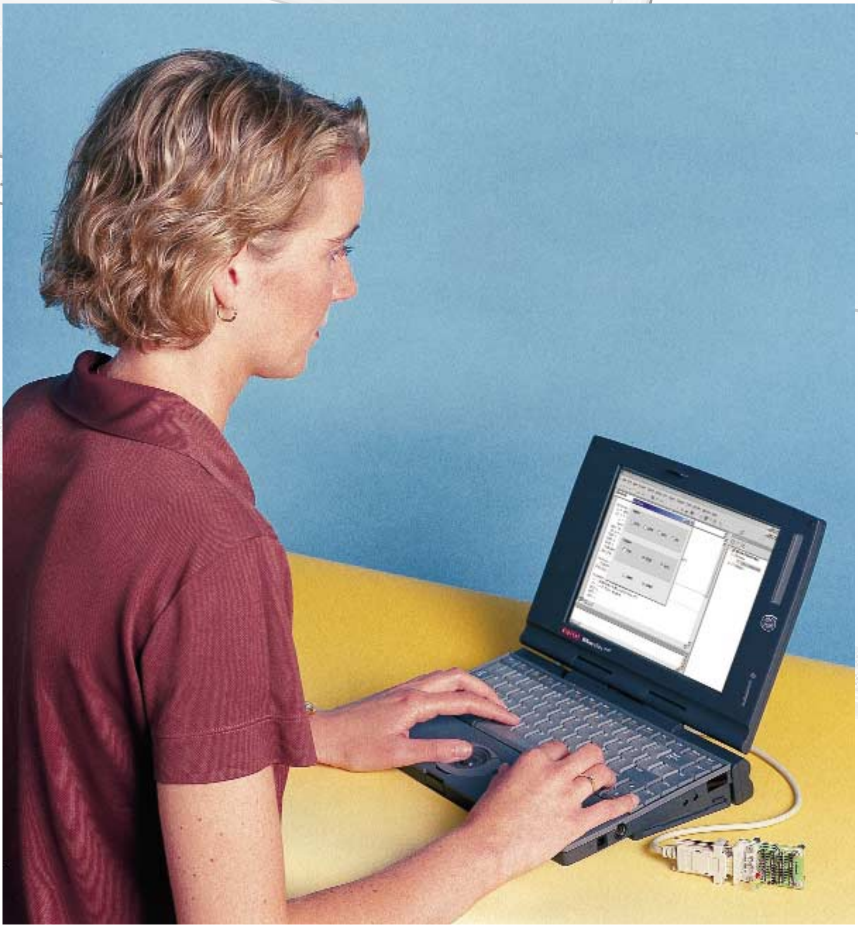
At one time or another many readers may have thought about controlling or monitoring equipment using their PC. Electronics projects using a PC need not be lavish or expensive: often the interfaces provided on the PC can be used directly. In this series of articles we present a range of projects using the serial (RS232) port, controlled using simple programs written in Visual Basic.

Why have we chosen to use the serial port, when it is more complicated and offers fewer connections than the printer port? For a beginners' course using the serial port has several advantages:

- The serial port is well protected against accidental damage. Cables can safely be plugged in while the computer is on.
- There is usually a spare serial port which can be used for experiments.
- The serial port can deliver enough current to power a wide range of projects, and so a separate power supply is not required.

This course is also easy on the pocket, which is important, not least for educational establishments and those on youth employment programmes. Apart from the small printed circuit board, you will need just a few everyday components, such as pushbuttons, NPN transistors, resistors, LEDs, capacitors, diodes and an LDR. All the connections of the serial port are brought out to sockets on the circuit board and there is also a small prototyping area for building experimental circuits.

All the projects are programmed in **Visual BASIC 5**. The programs (available on floppy disk, order code **000074-11**, or for download from *www.elektor-electronics.co.uk*) are clearly commented, so that you can easily



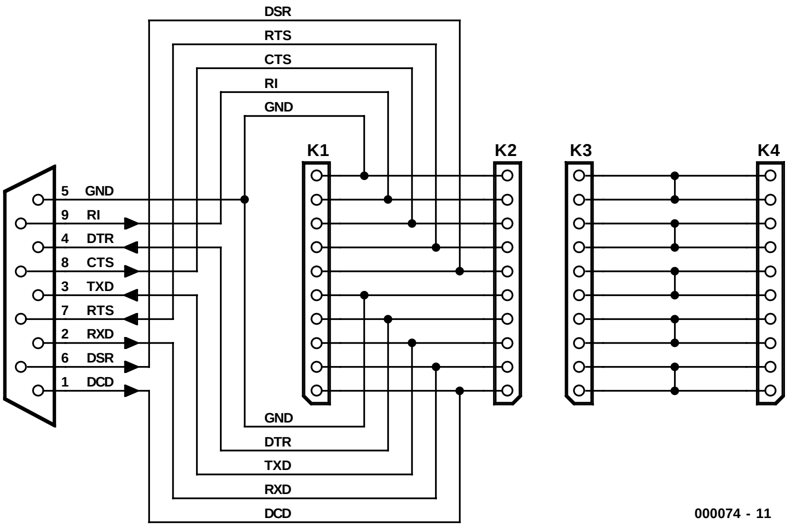


Figure 1. Circuit diagram of the experimental PCB.

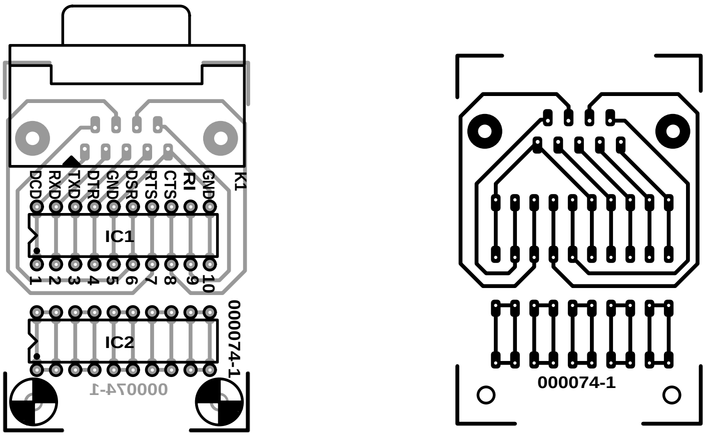


Figure 2. Component layout for the experimental PCB.

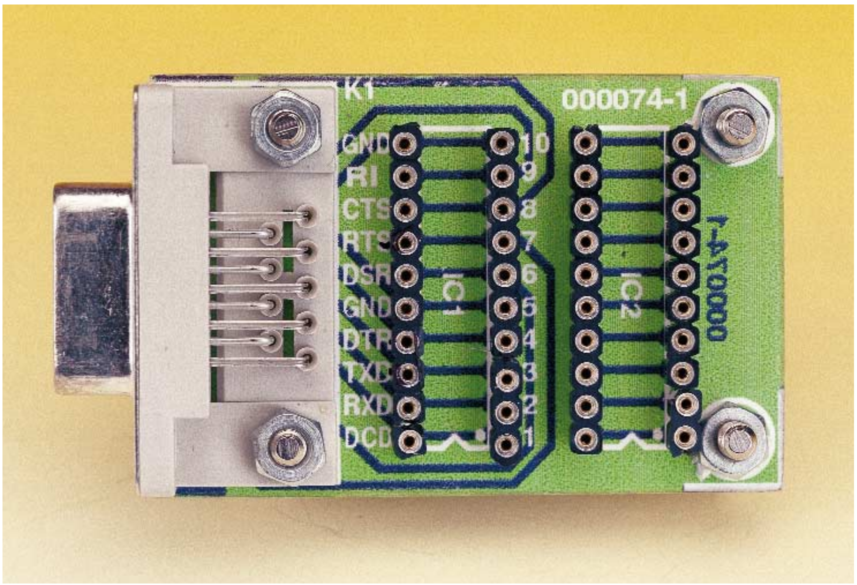


Figure 3. The assembled board.

change them to try out your own ideas. In later instalments, we will deal with complex topics such as real-time control and connection to external circuitry. As a bonus, readers will be introduced to applications of various sensor technologies, which arise naturally through the course.

The printed circuit board

The projects are based on a simple circuit board. As the circuit in **Figure 1** shows, it is a small prototyping board fitted with four SIL socket strips and a 9-way sub-D socket. Two of the four SIL sockets are connected to the sub-D socket, while the other two are simply connected to one another.

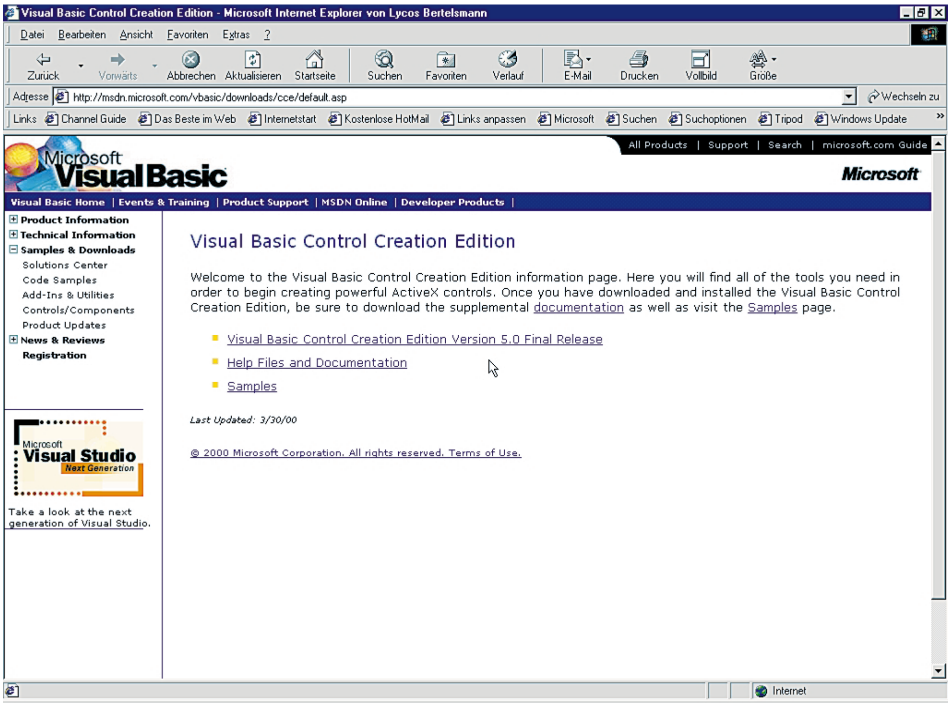
Figure 2 shows the component layout for the circuit board, and the assembled board is shown in **Figure 3**. **Table 1** gives a summary of the pinout of the 25-way and 9-way connectors normally employed for serial ports, showing the names and functions of all the signals that make up the interface. A male connector is invariably provided on the PC side, and so we require a female connector: a 9-way extension cable can be used to connect the PC to the circuit board described here. If your PC is equipped with a 25-way connector, you will need to use a suitable adapter.

Data is normally transferred over the serial interface using the TXD (transmit data) and RXD (receive data) signals. The other signals have auxiliary functions concerned with setting up and controlling data transfer. They are known as 'handshake signals', because they are used to acknowledge transfer of data between two pieces of equipment. A particularly useful feature of the handshake signals is that their state can be read or written directly.

The pin labelling on the circuit board follows that of the 9-way sub-D plug. Each pin of this plug is taken to two SIL socket pins, except for GND, which is taken to four. The small prototyping area on the circuit board is divided into five groups of connections with four SIL socket pins in each. Four SIL sockets with turned pins are used in total; two 20-pin DIL sockets — which are easier to obtain — can be used instead.

Visual BASIC

You will need a copy of Visual Basic, version 5 or 6, for this course. All source code is available from the *Elektor Electronics* website in VB5 format. The programs can also be loaded and compiled without modification using VB6. If you do not have a copy of Visual Basic



http://msdn.microsoft.com/vbasic/downloads/cce/default.asp

Visual Basic Control Creation Edition

Welcome to the Visual Basic Control Creation Edition information page. Here you will find all of the tools you need in order to begin creating powerful ActiveX controls. Once you have downloaded and installed the Visual Basic Control Creation Edition, be sure to download the supplemental documentation as well as visit the Samples page.

- Visual Basic Control Creation Edition Version 5.0 Final Release
- Help Files and Documentation
- Samples

Last Updated: 3/30/00

© 2000 Microsoft Corporation. All rights reserved. Terms of Use.

Figure 4. Visual Basic (VB5CCE) is available for download here.

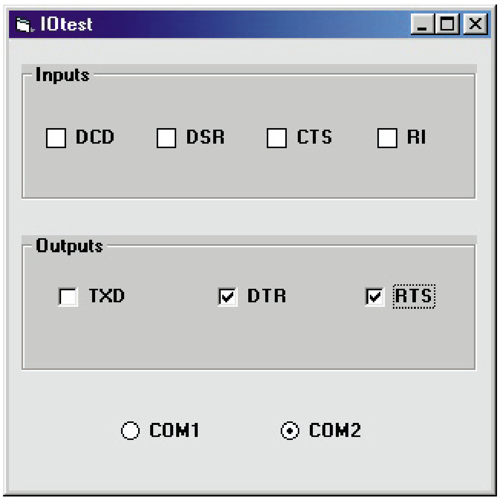
and balk at its price, you can obtain a free version of VB5 from Microsoft: 'Visual Basic Control Creation' VB5CCE is available on the Internet at

http://msdn.microsoft.com/vbasic/downloads/cce/default.asp

Our first program: I/O test

And now, at last, to our first program. It's a simple test program which gives direct access to all serial port connections (except RXD). The three outputs can be toggled with a click of the mouse, while the four inputs are read and displayed. The program is configured to use COM2 by default; however, it will recognise when COM2 is in use and automatically attempt to use COM1 instead. It is assumed that most PCs have a mouse connected to COM1 and that COM2 is unused. Frequently the connectors on PCs are not correctly labelled: in that case this program can help to identify them correctly. If you have two spare COM ports, you can use COM1 or COM2 as you prefer.

What is 'ON', and what is 'OFF'? If you connect a voltmeter between DTR and GND (ground), you will find that 'ON' gives a positive voltage of about 10 V, while 'OFF' gives a voltage of about –10V with respect to GND. Now connect a wire between, for example, the DTR output and the DSR input. As soon as DTR is turned



Inputs

☐ DCD ☐ DSR ☐ CTS ☐ RI

Outputs

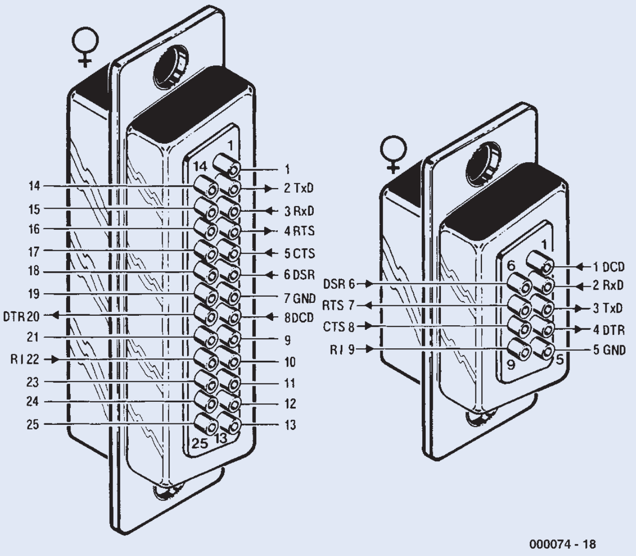
☐ TXD ☑ DTR ☑ RTS

○ COM1 ● COM2

Figure 5. Window showing direct access to serial port signals.

Table I Connector pinouts and signal names

Pin 25-way	Pin 9-way	Input/ Output	Symbol	Function
2	3	Out	TxD	Transmit Data
3	2	In	RxD	Receive Data
4	7	Out	RTS	Request To Send
5	8	In	CTS	Clear To Send
6	6	In	DSR	Data Set Ready
7	5		GND	Ground
8	1	In	DCD	Data Carrier Detect
20	4	Out	DTR	Data Terminal Ready
22	9	In	RI	Ring Indicator



on, the DSR input reports 'ON' too. Our first practical application is to poll a switch (**Figure 6**). In order to read the state of the switch, the DTR signal must be turned on. Now we have a complete PC application: the switch can be placed in some remote location and it state will be relayed to the PC. What that might represent is up to you: you might, perhaps, want to monitor whether that mouse-trap in the cellar has been set off.

We round off this first instalment with another neat little application. Here a light-emitting diode is connected to and controlled by the PC (**Figure 7**). Many *Elektor Electronics* readers will protest loudly that one should never connect an LED without a current-limiting resistor. Here, though, it is not necessary, as a resistor is effectively built into the PC.

Another protest: in the datasheets, the maximum allowable reverse voltage of an LED is given as 3 V or 5 V. In this circuit, however, reverse voltages of up to 10 V can appear across the LED. Experience has shown that this is not in practice a problem. Even with a reverse voltage of 20 V no significant current flows in most LEDs. But we should provide a 'clean' solution: an additional silicon diode (for example, a 1N4148) will protect the LED from excessive reverse voltages. The diode can be connected either in anti-parallel or in series with the LED (**Figure 8**). As a rule of thumb, for serious projects the reverse voltage should be limited to 5 V; for experimental purposes we can allow ourselves a little latitude.

(000074-1)

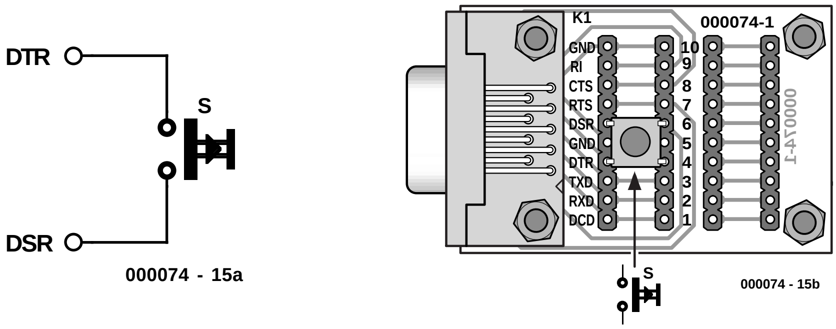


Figure 6. A simple switch is connected for the first experiment.

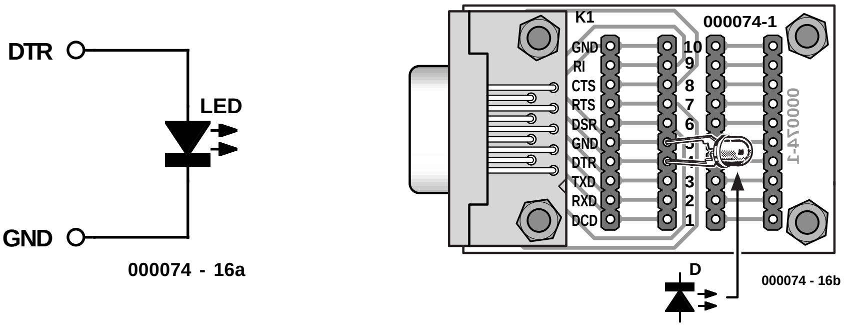


Figure 7. Equally simple: an LED is connected for the second project.

On Project Disk #000074-II:

PORTS.BAS	845	03-28-00	5:31p
ANPEL.FRX	1,094	04-10-00	6:01p
BLINK.FRX	1,094	04-10-00	6:43p
COUNTER1.FRX	1,094	05-04-00	6:49p
COUNTER2.FRX	1,094	05-04-00	7:14p
CONTENTS.TXT	0	06-15-00	9:35a
IOTEST.FRX	1,094	05-04-00	12:21p
AMPEL.VBP	497	04-10-00	6:01p
BLINK.VBP	497	04-10-00	6:43p
COUNTER1.VBP	499	05-04-00	6:49p
COUNTER2.VBP	499	05-04-00	7:14p
IOTEST.VBP	497	05-04-00	1:13p
AMPEL.VBW	81	05-05-00	12:57p
BLINK.VBW	80	05-05-00	4:26p
COUNTER2.VBW	80	05-05-00	4:27p
IOTEST.BW	80	06-08-00	1:15p
PORT.DLL	46,080	02-07-99	1:15p
BLINK.FRM	3,488	04-10-00	6:43p
COUNTER1.FRM	3,422	05-04-00	6:49p
COUNTER2.FRM	6,683	05-04-00	7:14p
IOTEST.FRM	5,317	05-04-00	12:21p
ANPEL.FRM	3,669	04-10-00	6:01p
22 file(s)	77,784 bytes		

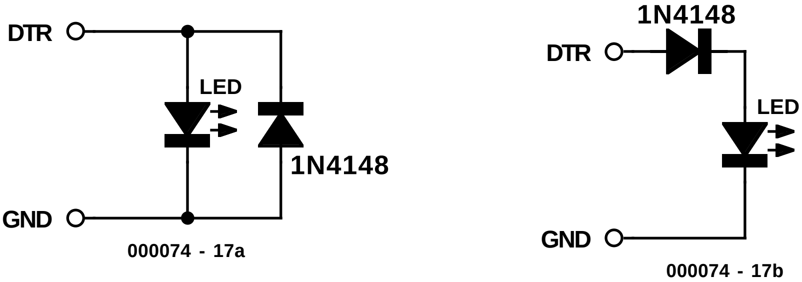


Figure 8. Two ways to protect the LED from excessive reverse voltage.